

A. AMENDMENTS TO CLAIMS

Please cancel Claims 3, 10 and 16 and amend the claims as indicated hereinafter.

- 1 1. (CURRENTLY AMENDED) A method for transforming character strings that are
2 contained in computer program source code the method comprising the computer-
3 implemented steps of:
4 automatically parsing the computer program source code to identify a hard coded
5 string that is both contained in the computer program source ~~code~~; code and
6 does not already have a corresponding macro string that is uniquely associated
7 with the hard coded string; and
8 in response to identifying a hard coded string that is both contained in the computer
9 program source code and does not already have a corresponding macro string
10 that is uniquely associated with the hard coded string,
11 replacing the hard coded string contained in the computer program source
12 code with a macro string that is uniquely associated with the hard
13 coded string;
14 creating and storing in a mapping of ~~macros~~ macro strings to hard coded
15 strings, an entry that defines an association between the hard coded
16 string and the macro string that replaced ~~and~~ the hard coded string; and
17 generating and storing in the computer program source code a reference to the
18 mapping of macros to strings.

- 1 2. (PREVIOUSLY PRESENTED) The method as recited in Claim 1, wherein the step
2 of automatically parsing the computer program source code to identify a hard coded
3 string includes:
4 identifying one or more computer program source code files that contain one or more
5 hard coded strings; and

6 automatically parsing at least one of the one or more computer program source code
7 files to identify the one or more hard coded strings while copying instructions
8 from at least one of the one or more computer program source code files to an
9 output.

1 3. (CANCELED)

1 4. (CURRENTLY AMENDED) The method as recited in claim 1, further comprising
2 the computer-implemented steps of:
3 receiving a suggested macro string for the identified hard coded string, and
4 generating the macro string to replace the hard coded string contained in the computer
5 program source code based upon the suggested ~~macro~~ macro string.

1 5. (PREVIOUSLY PRESENTED) The method as recited in claim 1, further comprising
2 the computer-implemented step of compiling the computer program source code to
3 generate an executable, including substituting in the executable the hard coded string
4 for each instance of the macro in the computer program source code.

1 6. (CURRENTLY AMENDED) The method as recited in Claim 1, further comprising
2 the computer-implemented steps of:
3 parsing the computer program source code to locate a second hard coded string
4 contained therein, wherein the second hard coded string is different than the
5 hard coded string;
6 in response to locating the second hard coded string contained in the computer
7 program source code, determining whether a macro string was previously
8 generated for the second hard coded string by searching the mapping; and

9 generating a second macro string uniquely associated with the second hard coded
10 string only when a macro string was not previously generated for the second
11 hard coded string.

1 7. (CURRENTLY AMENDED) A method for transforming hard coded character
2 strings that are contained in computer program source code the method comprising
3 the computer-implemented steps of:
4 identifying a hard coded string that is contained in the computer program source
5 code;
6 replacing the hard coded string contained in the computer program source code with a
7 macro string that is uniquely associated with the hard coded string;
8 creating and storing in a macro file a macro definition that defines an association
9 between the hard coded string and the macro string that replaced~~and~~ the hard
10 coded string; and
11 referencing the macro definition in the computer program source code using a
12 compiler directive that causes a compiler to include the macro file during
13 compilation of the computer program source code.

1 8. (CURRENTLY AMENDED) A computer-readable medium carrying one or more
2 sequences of instructions for transforming character strings that are contained in
3 computer program source code, wherein execution of the one or more sequences of
4 instructions by one or more processors causes the one or more processors to perform:
5 automatically parsing the computer program source code to identify a hard coded
6 string that is both contained in the computer program source ~~code~~; code and
7 does not already have a corresponding macro string that is uniquely associated
8 with the hard coded string; and

9 in response to identifying a hard coded string that is both contained in the computer
10 program source code and does not already have a corresponding macro string
11 that is uniquely associated with the hard coded string,
12 replacing the hard coded macro string contained in the computer program
13 source code with a macro that is uniquely associated with the hard
14 coded string;
15 creating and storing in a mapping of ~~macros~~ macro strings to hard coded
16 strings, an entry that defines an association between the hard coded
17 string and the macro string that replaced~~and~~ the hard coded string; and
18 generating and storing in the computer program source code a reference to the
19 mapping of macros to strings.

1 9. (PREVIOUSLY PRESENTED) The computer-readable medium as recited in Claim
2 8, wherein the step of
3 automatically parsing the computer program source code to identify a hard coded
4 string includes:
5 identifying one or more computer program source code files that contain one or more
6 hard coded strings; and
7 automatically parsing at least one of the one or more computer program source code
8 files to identify the one or more hard coded strings while copying instructions
9 from at least one of the one or more computer program source code files to an
10 output.

1 10. (CANCELED)

1 11. (CURRENTLY AMENDED) The computer-readable medium as recited in Claim 8,
2 further comprising the computer-implemented steps of:

3 receiving a suggested macro string for the identified hard coded string, and
4 generating the macro string to replace the hard coded string contained in the computer
5 program source code based upon the suggested ~~macro~~ macro string.

1 12. (PREVIOUSLY PRESENTED) The computer-readable medium as recited in Claim
2 8, further comprising the computer-implemented step of compiling the computer
3 program source code to generate an executable, including substituting in the
4 executable the hard coded string for each instance of the macro in the computer
5 program source code.

1 13. (CURRENTLY AMENDED) The computer-readable medium as recited in Claim 8,
2 further comprising the computer-implemented steps of:
3 parsing the computer program source code to locate a second hard coded string
4 contained therein, wherein the second hard coded string is different than the
5 hard coded string;
6 in response to locating a the second hard coded string contained in the computer
7 program source code, determining whether a macro string was previously
8 generated for the second hard coded string by searching the mapping; and
9 generating a second macro string uniquely associated with the second hard coded
10 string only when a macro string was not previously generated for the second
11 hard coded string.

1 14. (CURRENTLY AMENDED) A computer system for transforming character strings
2 that are contained in computer program source code stored in a memory, the computer
3 system comprising:
4 one or more processors coupled to the memory;

5 a conversion mechanism;

6 a stored mapping that defines one or more associations between macros and strings;

7 one or more computer instructions contained in the memory and associated with the

8 conversion mechanism which, when executed by the one or more processors,

9 cause the one or more processors to perform the steps of:

10 automatically parsing the computer program source code to identify a hard coded

11 string that is both contained in the computer program source ~~code~~; code and

12 does not already have a corresponding macro string that is uniquely associated

13 with the hard coded string; and

14 in response to identifying a hard coded string that is both contained in the

15 computer program source code and does not already have a

16 corresponding macro string that is uniquely associated with the hard

17 coded string,

18 replacing the hard coded string contained in the computer program

19 source code with a macro string that is uniquely associated

20 with the hard coded string;

21 creating and storing in the mapping that defines one or more

22 associations between ~~macros~~ macro strings and hard coded

23 strings, an entry that defines an association between the hard

24 coded string and the macro string that replaced~~and~~ the hard

25 coded string; and

26 generating and storing in the computer program source code a

27 reference to the mapping that defines one or more associations

28 between macros and strings.

1 15. (PREVIOUSLY PRESENTED) The computer system as recited in Claim 14,
2 wherein the step of automatically parsing the computer program source code to
3 identify a hard coded string includes:
4 identifying one or more computer program source code files that contain one or more
5 hard coded strings; and
6 automatically parsing at least one of the one or more computer program source code
7 files to identify the one or more hard coded strings while copying instructions
8 from at least one of the one or more computer program source code files to an
9 output.

1 16. (CANCELED)

1 17. (CURRENTLY AMENDED) The computer system as recited in Claim 14, further
2 comprising the computer-implemented steps of:
3 receiving a suggested macro string for the identified hard coded string, and
4 generating the macro string to replace the hard coded string contained in the computer
5 program source code based upon the suggested ~~macro~~ macro string.

1 18. (PREVIOUSLY PRESENTED) The computer system as recited in Claim 14, further
2 comprising the computer-implemented step of compiling the computer program
3 source code to generate an executable, including substituting in the executable the
4 hard coded string for each instance of the macro in the computer program source
5 code.

1 19. (CURRENTLY AMENDED) The computer system as recited in Claim 14, further
2 comprising the computer-implemented steps of:

3 parsing the computer program source code to locate a second hard coded string
4 contained therein, wherein the second hard coded string is different than the
5 hard coded string;
6 in response to locating the second hard coded string contained in the computer
7 program source code, determining whether a macro string was previously
8 generated for the second hard coded string by searching the mapping; and
9 generating a second macro string uniquely associated with the second hard coded
10 string only when a macro string was not previously generated for the second
11 hard coded string.

1 20. (CURRENTLY AMENDED) A computer-readable medium carrying one or more
2 sequences of instructions for transforming hard coded character strings that are
3 contained in computer program source code, wherein execution of the one or more
4 sequences of instructions by one or more processors causes the one or more
5 processors to perform the steps of:
6 identifying a hard coded string that is contained in the computer program source
7 code;
8 replacing the hard coded string in the computer program source code with a macro
9 string that is uniquely associated with the hard coded string;
10 creating and storing in a macro file a macro definition that defines an association of
11 between the hard coded string and the macro string that replaced~~and~~ the hard
12 coded string; and
13 referencing the macro definition in the computer program using a compiler directive
14 that causes a compiler to include the macro file during compilation of the
15 computer program source code.

1 21. (CURRENTLY AMENDED) An apparatus for transforming hard coded character
2 strings that are contained in computer program source code, the apparatus comprising
3 a memory carrying one or more sequences of instructions which, when executed by
4 one or more processors causes the one or more processors to perform the steps of:
5 identifying a hard coded string that is contained in the computer program source
6 code;
7 replacing the hard coded string contained in the computer program source code with a
8 macro string that is uniquely associated with the hard coded string;
9 creating and storing in a macro file a macro definition that defines an association
10 between the hard coded string and the macro string that replaced~~and~~ the hard
11 coded string; and
12 referencing the macro definition in the computer program source code using a
13 compiler directive that causes a compiler to include the macro file during
14 compilation of the computer program source code.